We claim:

1   1.   A virtual real time system for simulating a physical test environment comprising:

2   A master computer module; and

3   at least one slave computer module communicated to the master computer

4   module and having a clocked operation, which is synchronized to the master computer

5   module;

6   wherein the master computer module and at least one each slave computer

7   module each have a launcher submodule and a deployment submodule, the launcher

8   submodule for launching the deployment submodule and controlling the deployment

9   submodule for synchronized operation with the master computer module, the

10   deployment submodule generating a virtual clock and following commands from the

11   launcher submodule.

1   2.   The virtual real time system of claim 1 where the launcher submodule in the

2   master computer module is a central virtual real time controller for the system.

1   3.   The virtual real time system of claim 2 where the deployment submodule in the

2   master computer module generates a virtual clock signal based on process CPU

3   instruction execution.

1 4. The virtual real time system of claim 3 where the system includes an operating

2 system and where the launcher submodule and the deployment submodule in the

3 master computer module and in the slave computer module communicate via signals

4 provided by the operating system.

1 5. The virtual real time system of claim 1 further comprising a test master computer

2 submodule communicating with the master launcher submodule for configuring the

3 system and advancing, starting, stopping, adjusting and monitoring virtual real time,

4 and/or issuing time related commands to the deployment submodule in the master

5 computer module.

1 6. The virtual real time system of claim 5 where the master deployment submodule

2 generates a virtual clock signal and where test master computer submodule generates

3 scale-up and/or scale-down commands of the virtual clock in the master deployment

4 submodule.

1 7. The virtual real time system of claim 1 where the slave launcher submodule

2 further comprises a slave launcher synch submodule and where the slave launcher

3 submodule, upon receiving a command form the master launcher submodule, requests

4 the corresponding slave deployment submodule via the slave launcher synch

5 submodule to advance the slave deployment submodule by a predetermined number of

6 virtual clock ticks and to stop, after which the slave deployment submodule suspends

7 operation and waits for the slave launcher submodule to resume operation.

1   8.   The virtual real time system of claim 1 where master launcher submodule sends

2   a *start-tick* command to only to the slave launcher submodule, if it is prepared to receive

3   the next *start-tick* command by sending a socket call with a *start-tick* message.

1   9.   The virtual real time system of claim 1 where the slave deployment submodule

2   master deployment submodule each run and are included in a workstation, and where

3   the slave deployment is not running at the workstation where the master deployment

4   submodule is running.

1   10.   The virtual real time system of claim 7 where the slave launcher submodule after

2   receiving a *start-tick* command from the master deployment submodule sends a

3   SIGCONT signal to the suspended slave deployment submodule, the slave launcher

4   submodule sends an acknowledgment message to the master launcher submodule, the

5   slave deployment submodule in parallel with other programs runs the requested number

6   of ticks.

1   11.   The virtual real time system of claim 7 where the master launcher submodule

2   then sends a signal SIGCONT to its corresponding master deployment submodule to

3   run a requested number of virtual clock ticks based on Vclk clock ticks which are

4   generated when the time consumed by execution of process CPU instructions is equal

5   to or greater than tick-resolution time, the master deployment submodule suspends its

6   operation after running the requested number and the master launcher submodule waits

7   for the master deployment submodule to complete its cycles.

1   12.   The virtual real time system of claim 1 where the master launcher submodule

2   sends a *stop-tick* message to each slave launcher submodule which needs to be

3   synchronized at that clock tick based on slave tick synchronize size and a *stop-tick*

4   socket call is made to the candidate slave launcher submodule.

1   13.   The virtual real time system of claim 12 where the slave launcher submodule

2   after receiving a *stop-tick* command waits for a SIGSTOP signal from the slave

3   deployment submodule to make sure that the requested number of virtual clock ticks

4   has been completed, and the slave launcher submodule sends a *stop-tick*

5   *acknowledgment* message to the master launcher submodule.

1   14.   A method for operating a virtual real time system for simulating a physical test

2   environment comprising:

3   communicating at least one slave computer module with a master computer

4   module, which at least one slave computer module has a clocked operation and is

5   synchronized to the master computer module, wherein the master computer module

6   and at least one each slave computer module each have a launcher submodule and a

7   deployment submodule;

8   launching each of the deployment submodules corresponding to the launcher

9   submodules;

10   controlling each of the deployment submodules by the corresponding launcher

11   submodule for synchronized operation with the master computer module,

12   generating a virtual clock in the deployment submodule corresponding master

13   computer module; and

14   executing commands from the corresponding launcher submodule.

1   15.   The method of claim 14 further comprising providing a central virtual real time

2   controller for the system in the launcher submodule in the master computer module.

1   16.   The method of claim 15 where generating a virtual clock comprises generating a

2   virtual clock signal based on process CPU instruction execution in the deployment

3   submodule in the master computer module.

1   17.   The method of claim 16 where the system includes an operating system and

2   where communicating at least one slave computer module with a master computer

3   further comprises communicating between the launcher submodule and the deployment

4   submodule in the master computer module and in the slave computer module via

5   signals provided by the operating system.

1   18.   The method of claim 14 further comprising communicating with the master

2   launcher submodule with a test master computer submodule for configuring the system

3   and advancing, starting, stopping, adjusting and monitoring virtual real time, and/or

4   issuing time related commands to the deployment submodule in the master computer

5   module.

1   19.   The method of claim 18 where generating a virtual clock in the deployment

2   submodule corresponding master computer module comprises generating scale-up

3   and/or scale-down commands of the virtual clock in the master deployment submodule

4   by means of the test master computer submodule.

1   20.   The method of claim 14 where the slave launcher submodule further comprises a

2   slave launcher synch submodule and where the slave launcher submodule, upon

3   receiving a command form the master launcher submodule, further comprising

4   requesting the corresponding slave deployment submodule via the slave launcher

5   synch submodule to advance the slave deployment submodule by a predetermined

6   number of virtual clock ticks and to stop, after which the slave deployment submodule

7   suspends operation and waits for the slave launcher submodule to resume operation.

1   21.   The method of claim 14 further comprising sending a *start-tick* command to only

2   to the slave launcher submodule from the master launcher submodule, if the slave

3   launcher submodule is prepared to receive the next *start-tick* command by sending a

4   socket call with a *start-tick* message.

1   22.   The method of claim 14 further comprising running the slave deployment

2   submodule master deployment submodule each in a workstation, and running the slave

3   deployment at the workstation other than where the master deployment submodule is

4   running.

1   23.   The method of claim 20 further comprising sending a SIGCONT signal to the

2   suspended slave deployment submodule from the slave launcher submodule after

3   receiving a *start-tick* command from the master deployment submodule, sending an

4   acknowledgment message from the slave launcher submodule to the master launcher

5   submodule, and running the slave deployment submodule in parallel with other

6   programs the requested number of ticks.

1   24.   The method of claim 20 further comprising sending a signal SIGCONT from the

2   master launcher submodule to its corresponding master deployment submodule to run a

3   requested number of virtual clock ticks based on Vclk clock ticks which are generated

4   when the time consumed by execution of process CPU instructions is equal to or

5   greater than tick-resolution time, suspending operation of the master deployment

6   submodule after running the requested number, and forcing the master launcher

7   submodule to wait for the master deployment submodule to complete its cycles.

1   25.   The method of claim 14 further comprising sending a *stop-tick* message from the

2   master launcher submodule to each slave launcher submodule which needs to be

3   synchronized at that clock tick based on slave tick synchronize size and a *stop-tick*

4   socket call is made to the candidate slave launcher submodule.

1   26.   The method of claim 25 further comprising forcing the slave launcher submodule

2   to wait after receiving a *stop-tick* command for a SIGSTOP signal from the slave

3   deployment submodule to make sure that the requested number of virtual clock ticks

4   has been completed, and sending a *stop-tick acknowledgment* message from the slave

5   launcher submodule to the master launcher submodule.